MagicalHands: Mid-Air Hand Gestures for Animating in VR

Rahul Arora[1] **Rubaiat Habib Kazi**[2] **Danny Kaufman**[2] **Wilmot Li**[2] **Karan Singh**[1]

[1]University of Toronto
Toronto, Canada
{arorar,karan}@dgp.toronto.edu

[2]Adobe Research
Seattle, USA
{rhabib,kaufman,wilmotli}@adobe.com

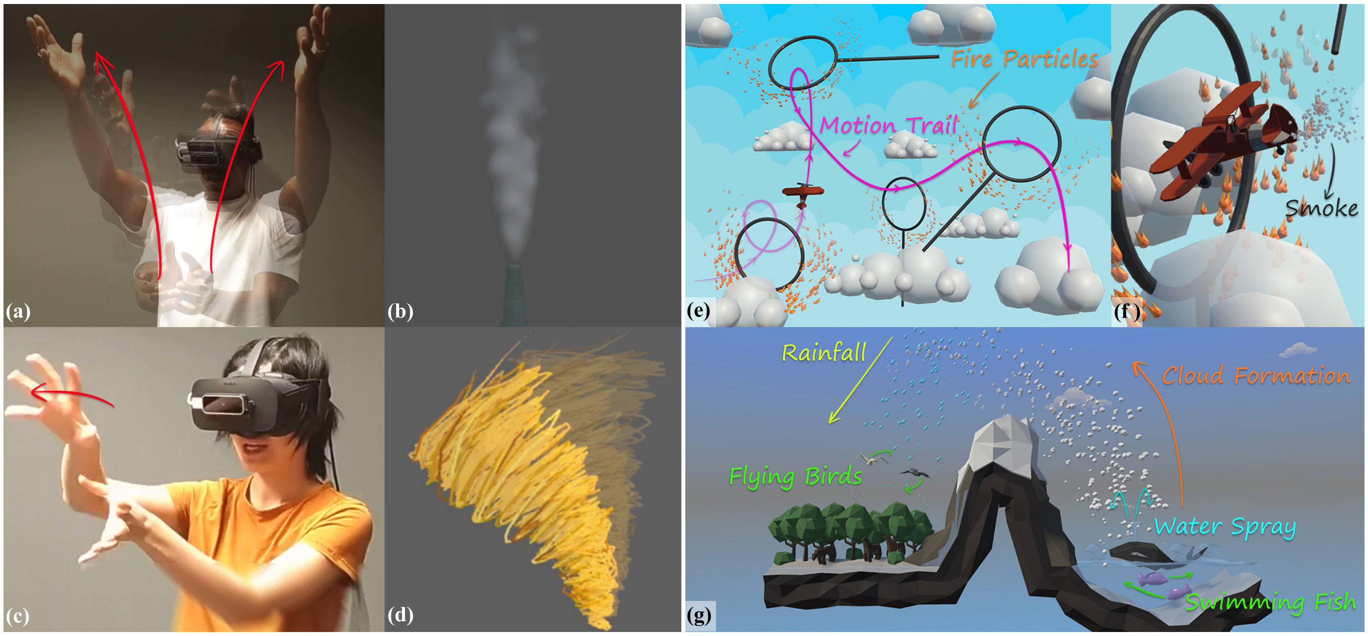


Figure 1. We studied the use of mid-air gestures for animation authoring in VR, utilizing high-level creation tasks to understand the basic operations utilized by animators, and the features of gestures undertaken to effectuate the same. Here, we show two example gestures from our study: a high-bandwidth gesture manipulating the direction, spread, and randomness of smoke emission (a–b) and a gesture directly bending an object to describe a *follow-through* behaviour (c–d). We then built an animation system—MagicalHands—based on the insights gained from the study. The system supports 3D manipulation and particle systems, which we used to create an airplane stunt scene (e–f) and a water cycle visualization (g). Annotations added to indicate animated phenomena. Please see the accompanying video for the effects in action. Water cycle model ©Hermes Alvarado; used with permission.

ABSTRACT

We explore the use of hand gestures for authoring animations in virtual reality (VR). We first perform a gesture elicitation study to understand user preferences for a spatiotemporal, bare-handed interaction system in VR. Specifically, we focus on creating and editing dynamic, physical phenomena (e.g., particle systems, deformations, coupling), where the mapping from gestures to animation is ambiguous and indirect. We present commonly observed mid-air gestures from the study that cover a wide range of interaction techniques, from direct manipulation to abstract demonstrations. To this end, we extend existing gesture taxonomies to the rich spatiotemporal interaction space of the target domain and distill our findings into a set of guidelines that inform the design of natural user interfaces for VR animation. Finally, based on our guidelines, we develop a proof-of-concept gesture-based VR animation system, MagicalHands. Our results, as well as feedback from user evaluation, suggest that the expressive qualities of hand gestures help users animate more effectively in VR.

UIST '19, October 20-23, 2019, New Orleans, LA, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6816-2/19/10 ...$15.00.
http://dx.doi.org/10.1145/3332165.3347942

Author Keywords

hand gestures; animation; gesture elicitation; virtual reality

CCS Concepts

•Human-centered computing → **Gestural input;** *Virtual reality;* Participatory design; **•Computing methodologies** → **Animation;**

INTRODUCTION

Hand gestures are a ubiquitous tool for human-to-human communication, often employed in conjunction with or as an alternative to verbal interaction. They are the physical expression of mental concepts [50], thus augmenting our communication capabilities beyond speech [31]. The expressive power of gestures has inspired a large body of human-computer interaction research on gesture-based interfaces, from generic 2D manipulation techniques [39, 57, 58] to specialized tools for 3D modelling [43, 65, 27], 3D model retrieval [18, 64], and visualization [4]. Mid-air gestures allow users to form hand shapes they would naturally use when interacting with real-world objects [64, 47], thus exploiting users' knowledge and experience with physical devices.

Our work investigates the application of gestures to the emerging domain of VR animation [14, 33, 54, 34], a compelling new medium whose immersive nature provides a natural setting for gesture-based authoring. Current VR animation tools are primarily dominated by controller-based direct manipulation techniques, and do not fully leverage the breadth of interaction techniques and expressiveness afforded by hand gestures. We explore how gestural interactions can be used to specify and control various spatial and temporal properties of dynamic, physical phenomena in VR.

A key contribution of our work is an empirical study that explores user preferences of mid-air gestures for animation. The main challenge for such a study is that animation is a vast discipline with distinct and diverse authoring strategies [36]. Moreover, even animating a simple scene can be a complex task with multiple workflows and many individual operations. Thus, unlike previous gesture elicitation studies [60, 41], we cannot easily define and collect gestures for a pre-defined set of atomic target operations.

To address this challenge, we restrict the scope of our investigation to physics-based animation. The physical plausibility of effects like collisions, deformations, and particle systems can strongly enhance immersion and presence in VR, which makes this an important class of phenomena to study. In addition, unlike prior work on performance-based animation that leverages the relatively direct mapping between humans and articulated digital characters [37, 62], the mapping between human gestures physics-based effects is much more ambiguous and requires further investigation.

In terms of the study design, we developed a methodology that supports the unstructured nature of animation tasks. We created several simple but realistic animated scenes in VR to act as the target stimuli. We recruited 12 professional animators and asked them to create each animation (which involved multiple individual steps) using gestures, while following a think aloud protocol. Based on our observations and their spoken explanations, we segmented the data into individual interactions that were then analyzed along several dimensions. This approach allowed us to compare the use of gestures across participants and identify common usage patterns.

Our study elicited many high-bandwidth gestures assuming simultaneous control of a number of parameters (Figure 1). However, we also found that most gestures did not encode information in the nuances of hand shape, and thus, even coarse hand-pose recognition should suffice for building gestural VR animation interfaces. From our findings, we propose several design guidelines for future, gesture-based, VR animation authoring tools. Finally, as a proof of concept, we implemented a system, MagicalHands, consisting of 11 of our most commonly observed interaction techniques. Our implementation leverages hand pose information to perform direct manipulation and abstract demonstrations for animating in VR.

To summarize, our main contributions are

- a gesture elicitation study on creating dynamic phenomena in VR,
- a taxonomy of mid-air gestures for VR animation,
- a set of commonly observed gestures for VR animation,
- a set of design guidelines for VR animation tools, and
- a prototype animation system and informal study to assess the efficacy of gestural interactions for VR animation.

RELATED WORK

Our work relates to literature in gesture elicitation, expressiveness of gestural inputs, and animation interfaces.

Gesture Elicitation and Classification

Traditionally, gestural interfaces have been designed by experts and tested a posteriori [39, 61]. The contrasting interface design methodology is *participatory design*, where users participate in the formative design process. Wobbrock et al. [60] first advocated this process for gesture design, with an elicitation study for performing canonical UI manipulation and navigation tasks on the table-top. Following Wobbrock et al., *user-designed gestures* have been utilized to build interfaces for a variety of domains [41, 40, 56, 53]. Particularly relevant to our study are the recent works on grasping [64], remote object manipulation [66] and music composition [28] in VR environments, and the elicitation study of Piumsomboon et al. [38] for AR-based interfaces. We follow the same guiding philosophy for building a VR animation interface.

However, an important point of departure is that all these works assumed a well-defined set of atomic operations for eliciting related gestures, which is not desirable in our case. In this aspect, the closest exploration is that of Aigner et al. [3], which attempts to elicit gestures for complex multi-part tasks. Still, they focused on UI manipulation and navigation as well, and we are unaware of any gesture elicitation studies for authoring complex, spatio-temporal phenomena for animation.

Another key contribution of our work is a taxonomy of gestures useful for VR-based animation tasks. While we are not aware of any existing work exploring gesture taxonomies for animation authoring, we do build on prior gesture taxonomies proposed for other application scenarios. McNeill's well-known classification [31] partitions gestures into *iconics*, *metaphorics*, *beats*, *cohesives*, and *deictics*. However, such taxonomies proposed in linguistics and psychology focus on gestures executed to augment speech in human communication. We leverage more recent HCI work on gesture classification by Karam and schraefel [21], Wobbrock et al. [60] and Aigner et al. [3] and classify gestures based on the level of implied abstraction. But, unlike prior works, our classification scheme is customized for animation-specific spatiotemporal tasks.

Expressiveness of Gestural Inputs

Researchers have leveraged the communicative aspects of freeform hand gestures in computing systems for 3D model retrieval [18, 64], approximating 3D shapes for early stage design [27], and interacting with imaginary devices [47]. There is however, no empirical study to understand user performance and preference of mid-air gestures for animation tasks. Our paper contributes such a study, and a corresponding proof-of-concept gestural animation system based on the study.

From a mechanical perspective, the human hand has 27 DoFs [2]. As such, the number of independent parameters a mid-air gesture can simultaneously specify is theoretically very high (54 for bimanual). However, many of these DoFs are heavily intertwined and have severely limited ranges of independent motion [30]. Further, the number of parameters that can be simultaneously manipulated is restricted by users' mental models relating parameters to each other and by the cognitive load of the task [17]. Remi et al. [8] conducted a study to understand the *true* number of DoFs users specified in table-top gestures for 3D manipulation. Their observations are equally applicable for mid-air gestural inputs. While we don't quantitatively specify the number of degrees present in gestures for general mid-air manipulation tasks, we qualitatively observe user behaviour and suggest principles governing simultaneous parameter specification.

Natural Interfaces for Animation Authoring

Most professional animators use specialized tools with complicated interfaces full of sliders and numeric entries for controlling thousands of different commands and parameters. An alternative approach is *performance-driven* animation [37, 62, 1, 16], which tracks human actors and maps their motion to digital characters. Over the last decade, researchers have also explored *direct manipulation* interfaces to make animation easy and accessible [23, 22, 52, 12, 19, 63, 24], allowing amateur users to rapidly adapt to their animation creation tools.

Spatial and gesture-based interfaces have been defined for more restricted animation tasks. For instance, Finger Walking [29] to authoring walking/running motion, motion capture widget [13] to control a restricted set of DoFs of a biped or quadruped character, full-body motion [9] to non-humanoid characters with various topologies, and finger movements [32] to deformable, rigged drawings. While restricted to specific sub-domains of animation, these works show that gestures and spatial inputs can be used to simultaneously control multiple DoFs in animated scenes. However, in these cases, the mapping between the input and animation parameters is natural and well understood. Jensen et al. [20] explored how to extract physical properties of virtual objects from demonstrations.

Commercial VR animations tools, such as Quill [14] and Tvori [54], apply performance-based direct manipulation to animated strokes and articulated skeletons using hand-held controllers. In contrast to direct manipulation, the communicative aspects of hand gestures to articulate dynamic, physical phenomena remains an open and challenging question. In computer graphics, the development of algorithms to control and guide physical simulations—such as deformation, collision, flow, and fracture—is a long-standing area of research [7, 35, 6]. We investigate the mapping between spatial interactions (hand gestures) and physical animation parameters, and provide guidelines for future spatial animation interfaces.

HAND GESTURE USAGE STUDY

Our overall goal was to elicit a broad range of gestures for creating VR animations in order to derive design guidelines for a gestural animation system. A challenge of this setting is that animation involves many individual operations to select and modify various spatial and temporal properties. Furthermore, animators can choose to specify such properties at the individual entity level or control high-level physical (e.g., material stiffness) or abstract (e.g., amount of noise) parameters. Thus, a key objective for our study was to elicit both a diverse set of atomic operations and various gestures for executing those tasks. To this end, we created a range of target scenes covering common dynamic effects and then asked participants to describe operations and gestures to create each animation.

Target Animated Scenes

Animated stories include a diverse range of motions and dynamic effects. As noted earlier, previous research on gesture-based animation largely focuses on the problem of animating humanoid characters through facial or full-body performance where the degrees of freedom of the performer and the target character are often very similar. Our study focuses on common classes of physical systems like particles, fluids, and multi-body interactions where the gesture-to-motion mapping is less obvious than for characters.

Based on our analysis of animation literature [36, 63, 15] and physical phenomena featured in commercial animation software [45, 49, 5], we created six scenes covering a range of physical effects (Figure 2). For each scene, we generated 3–4 target animations to cover typical stylistic variations and presented these variants to participants in increasing order of visual complexity.

Bouncing Ball. This scene shows a ball bouncing once on the ground. It includes a collision and a response, typical for physically-based object interactions. The first animation shows the ball moving at a constant speed; the second exhibits easing around the collision event; and in the third, the ball squashes and stretches. This scene thus helps study rigid-body interaction as well as elastic deformation.

Tornado. This scene involves layering multiple simultaneous motions on an object. The first animation shows a rotating tornado; the second adds a motion path; and the third includes a follow-through effect, bending at the end of the path.

Bubbles. This scene includes variations of a simple particle system where bubbles emanate from a wand. In the first animation, the bubbles appear at regular intervals in groups of 10; the second adds high-frequency vibrations to each bubble; and the third staggers the timing so that bubbles appear one at a time. This scene was included to understand how animators gesturally interact with a particle system, an entity typically controlled using multitudes of abstract parameters.

Hook and Spring. This scene investigates physical coupling and decoupling of objects, which occurs in many multi-body interactions. Animators often use such interactions to emphasize the physical attributes of objects and add secondary motion to a scene. The initial animation involves a hook picking up a spring object as it moves along a path; the second adds vertical vibrations to the spring after it gets picked up; and in the third, the hook rotates to drop the spring at the end.

Sheet. This scene includes shape and topology changes to a sheet that gets stretched from opposite edges. The first animation shows the sheet stretching in an area-preserving manner; in the second, the sheet tears apart down the middle; the third variant modifies the shape of the tear to be a complex curve; and the last shows the sheet disintegrating into small pieces, which fly away as it tears. These animations explore the well-studied cloth simulation technique as well as the topological changes introduced by ripping and shattering.

Smoke. The last scene involves several variations of a smoke simulation. The first variant shows fairly laminar flow straight up from a chimney; in the second, the smoke follows a curved, non-planar path; finally, the last clip adds turbulent noise and dispersion to the flow. This scene was utilized to understand how gestural control of a continuum of fluid can be achieved.

Of course, these examples do not cover all possible physical phenomena. However, we believe our target animations (19 in total) span a wide range of dynamic effects and modifications that are common across real-world animations.

Participants

We recruited 12 animators (9M, 3F) aged 19 to 58 (median 42). All participants had a minimum of two years of experience creating animations, and five had over a decade of experience. Participants had diverse backgrounds in 2D animation, 3D character animation, motion graphics, and procedural animation. Participants had experience with numerous software packages including After Effects, Maya, Blender, Character Animator, Apple Motion, Toon Boom Harmony and Unity. While most had tried VR, none had authored VR-animations.

For four animators (P1–P4), we conducted an initial session with the first three target scenes. P1 and P2 completed the remaining target scenes in a follow-up session, but P3 and P4 were unavailable. The other participants (P5–P12) went through the study in one session with a break in the middle. The study took between 75 and 120 minutes.

Apparatus and Implementation

Participants wore an Oculus Rift HMD with a Leap Motion hand tracker mounted on it, allowing participants to see non-photorealistic renderings of their hands while gesturing (Figure 3). Note that allowing participants to see their hands was the sole purpose of the Leap Motion device, and the tracking data was not utilized for analysis.

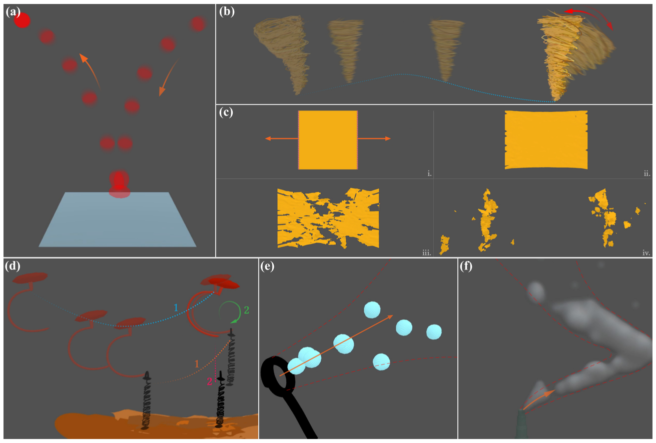


Figure 2. The six phenomena studied in our experiment—showing one clip from each phenomenon here. Bouncing ball with squash and stretch (a), tornado motion with follow through/bending (b), topological changes in shattering sheet animation (c), physical coupling and decoupling (d), particle system making a bubble blowing animation (e), and smoke simulation (f). Please see the supplementary video for all the animated clips.

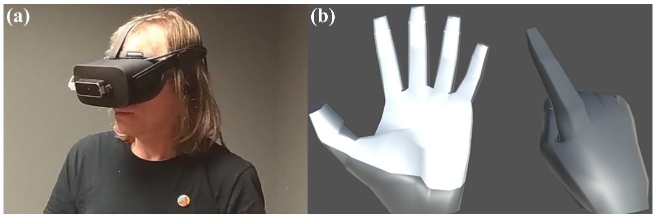


Figure 3. Experimental setup: Leap Motion mounted on an Oculus HMD (a), enabling participants to see their hands while gesturing (b).

The first four target scenes were created using Oculus Quill's frame-by-frame animation tools. *Sheet* animations were simulated in Maya [5] and then imported into Unity [55], while the *smoke* was simulated in real-time using the smoothed-particle hydrodynamics (SPH) [25] implementation in Fluvio [51].

Procedure

In each session, we presented target animations to the participant in the order listed above. For each example, we started by showing the static scene objects and then playing the target animation. We told participants to consider the static objects as the input for the task and author the animation with gestures. Since our aim was to elicit a broad range of behaviors, we encouraged participants to demonstrate additional gestures after their initial attempt for each task. Moreover, we expected the initial interaction to be biased by the dominant interface elements in participants' preferred animation software, and eliciting additional gestures could help alleviate this bias.

An experimenter was always present to answer questions, provide clarifications, and update the state of the virtual scene (e.g., participants could ask to see the static objects, replay the animation, or pause at a frame). Sessions were video-recorded for later analysis.

During each task, we asked participants to think-aloud as they worked. Given the complexity of our target tasks, it was critical for participants to verbalize the intent of each gesture and describe how they expected the system to respond. For example, in some cases, participants demonstrated high bandwidth gestures and explained how these actions were meant to accomplish multiple atomic operations. In other situations, participants proposed the use of traditional UI components like menus, buttons, and sliders to modify specific attributes of the animation. Here, we were aided by our choice of professional animators as participants: lacking direct visual feedback from the scene, novices may have missed controlling some of the more subtle aspects of the complex and diverse animation tasks. In this respect, our experiment differs considerably from more structured elicitation studies such as Wobbrock's [60], where each participant's gestures have a clear one-to-one mapping with the intended effects.

RESULTS

We analyzed the recorded user sessions to identify the most commonly observed gestures for our VR animation tasks (Fig. 4). We further analyzed the gestural interactions by taxonomizing along geometric and semantic dimensions and identifying common trends in how participants expressed various spatio-temporal operations.

Analysis Methodology

For each participant, we manually segmented the recording into separate disjoint interactions. Most interactions (493 total) were gestural, and we categorized these based on existing gesture taxonomies. Since participants also had the option of describing interactions with traditional widgets, we also observed 133 non-gestural interactions with imaginary menus, buttons, sliders, etc. Fig.4 presents commonly observed gestures during our study sessions for transformations, deformations, coupling, and interacting with particle systems. The resulting gestures demonstrate the breadth and richness of mid-air interaction techniques to control dynamic phenomena.

Taxonomies of Interactions

We categorized the gestural interactions along four separate dimensions: two that characterize geometric features, and two that describe the semantics of all interactions. For each gesture, we also categorized the effect participants wanted to execute with the interaction. After forming the list of codes, two authors independently coded each interaction for a subset of data (approx. 10%) into the taxonomy of actions and effects (Table 2) and as either gestural or non-gestural, with gestural interactions further coded into a taxonomy of gestures (Table 1). Since the Cohen's kappa measure showed excellent agreement between the two authors on all four dimensions ($\kappa \in [0.81, 0.97]$), one author proceeded to code the rest of the data all by themselves.

Hand Usage	*unimanual*	Only one hand is actively utilized.
	bimanual-static	One hand moves, the other is fixed.
	bimanual-symmetric	Both hands move symmetrically.
	bimanual-other	Both hands move independently.
Form	*static pose*	Hand pose is held at one location.
	dynamic pose	Hand pose changes at one location.
	static pose and path	Hand pose is held while it moves.
	dynamic pose and path	Hand pose changes as it moves.

Table 1. Taxonomy of mid-air gestures for animation.

Geometric Taxonomy of Gestures

We classified gestures along taxonomies based on hand usage [8] and form [60] (Table 1).

Hand usage distinguishes between *unimanual* and *bimanual* gestures, with bimanual interactions further classified as *static* where one hand stays still, *symmetric* if both hands move about a point or plane of symmetry, or *other* if the two hands move independently. Typical examples include deforming an object by using one hand as a static constraint (Fig. 1c–d: *static*), defining an emission cone by tracing its silhouette (Fig. 4d: *symmetric*), and moving in time with one hand while setting a parameter with the second (*other*).

Form categorizes if and how the pose and position of hands vary within a gesture. Table 1 briefly describes and Figure 5 illustrates the four categories. See Wobbrock et al. [60] for details. Examples include using extended hands to pause: *static pose*; tracing out a path (Fig. 4a): *static pose and path*; tapping to select (Fig. 4h): *dynamic pose*; and throwing objects (Fig. 4a): *dynamic pose and path*. A bimanual gesture is considered to have a dynamic pose and/or path if either of the hands satisfies the respective criteria.

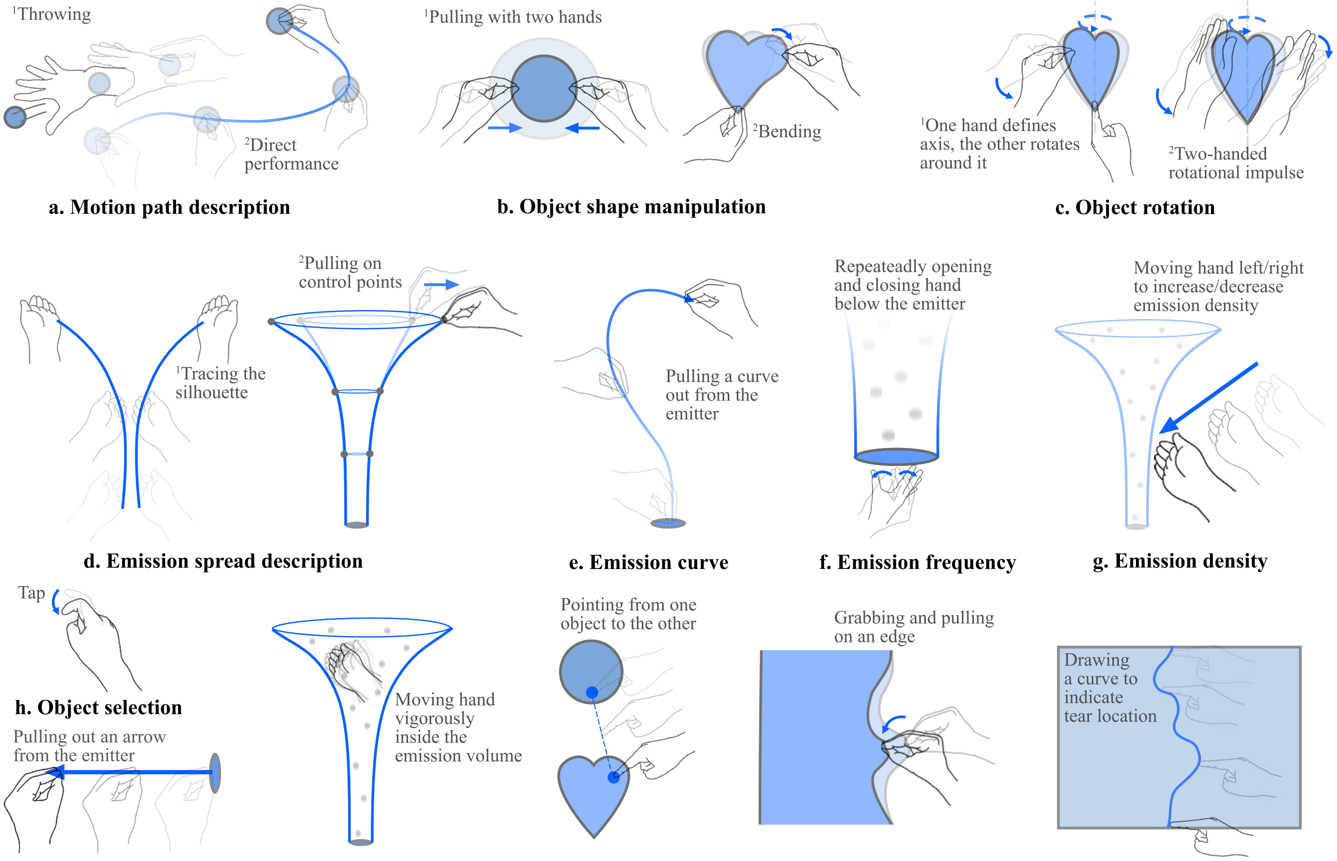
Semantic Taxonomy of Interactions

For all gestural and non-gestural interactions, we also considered what desired action the interaction conveys (i.e., the *nature of action*) and the intended effect on the animation (i.e., the *nature of effect*).

Nature of Action	
direct manipulation	Hand(s) interact with objects in the scene directly.
demonstrative	Features of hand pose and/or path emulate the intended effect.
semaphoric	Abstract or learned relation between gesture and effect.
widget use	Manipulation via a traditional widget, such as a push button, toggle, or slider.
Nature of Effect	
spatial	Addition or editing of a purely spatial property.
temporal	Manipulation of the timing of an animated effect.
spatiotemporal	Altering both spatial and temporal properties.
abstract	Manipulation at a higher abstraction level.
interface	Navigational tools that do not modify the animation.

Table 2. Taxonomy of interactions based on the nature of the user's action and the intended effect for VR-based animation authoring.

To classify the *nature of actions*, we took cues from existing gesture taxonomies [21, 60, 3], but customized the categories to focus on animation-specific features (Table 2). *Direct Manipulation* gestures directly influence an object in the scene, including direct performances of trajectories, sculpting to alter object shape, and throwing or flicking gestures that induce motion via an implied underlying physics engine. In contrast, *demonstrative* gestures do not manipulate objects directly, but some spatiotemporal properties of the action mimic the desired effect. Examples include mimicking the shape change of an object using the hand's pose, manipulation of assumed spatial interfaces such as emission cones or motion trails, and *lasso*-ing to group objects. *Semaphoric* gestures—such as extending both hands forward to indicate pause or stop—rely on accepted interpretations of specific gestures or poses to indicate a desired effect. Finally, users wanted to perform some operations using abstract controls such as buttons and



[1]Throwing
[2]Direct performance
[1]Pulling with two hands
[2]Bending
[1]One hand defines axis, the other rotates around it
[2]Two-handed rotational impulse
[2]Pulling on control points
[1]Tracing the silhouette
Pulling a curve out from the emitter
Repeateadly opening and closing hand below the emitter
Moving hand left/right to increase/decrease emission density
Tap
Pulling out an arrow from the emitter
Moving hand vigorously inside the emission volume
Pointing from one object to the other
Grabbing and pulling on an edge
Drawing a curve to indicate tear location

a. Motion path description

b. Object shape manipulation

c. Object rotation

d. Emission spread description

e. Emission curve

f. Emission frequency

g. Emission density

h. Object selection

i. Emission speed

j. Randomness description

k. Coupling

l. Local shape manipulation

m. Topological change (spatial)

Figure 4. Most commonly observed gestures (dominant gestures for commands which had at least 10 occurences in our study). These gestures include direct manipulation (a–c) and abstract demonstrations (d, f, g, i, j). Many involve interactions with physics (a, c), simultaneously specifying multiple parameters, such as speed and emission cone (d), and coordinating multiple parallel tasks using bi-manual interactions, where the non-dominant hand specifies the rotation axis, and the dominant hand records the motion (d)

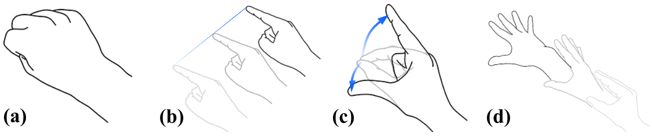


Figure 5. Gesture *form* classification: *static pose* (a), *static pose and path* (b), *dynamic pose* (c), and *dynamic pose and path*.

sliders. Tasks where such *widget uses* dominate may not be well-suited for gestural control.

We also considered the *nature of effects* that each interaction intended to produce. These categories are specific to the animation authoring domain. Some interactions only modified *spatial* properties of the scene, such as scaling or positioning an object at a given instant to set a keyframe. Conversely, some interactions only modified *temporal* properties, such as stretching a motion trail like an elastic band to locally "expand time", or non-gestural interactions to set speed variables. Other interactions modified *spatiotemporal* attributes. For example, direct demonstration of how a spatial property changes over time, such as tracing an object's rotation or pulling away from an emitter to describe both the speed and trajectory of particles. Throwing and flicking manipulations also fall into this category since they induce spatiotemporal behavior via implied physics. *Abstract* interactions such as duplication, coupling objects together, and setting physical parameters (e.g., mass, elasticity) manipulate the animation at a more abstract level, beyond explicit spatial or temporal scene properties. Finally, *interface* interactions were used to navigate through space and/or time (e.g., to view a specific part of the clip) and did not explicitly modify the animation.

Taxonomic Distributions and Relationships

Fig. 6 shows how the elicited interactions were distributed across the classification dimensions. Here, we note interesting second-order effects between the classification dimensions.

A large number (35%) of interactions were *direct manipulations* (Fig. 4a-c), which can perhaps be explained by the enhanced feeling of presence attributed by participants (P3, 4, 7, 10–12) to the immersive space and hand tracking. Several of these gestures included participants "performing" the intended motion, illustrated by two-thirds of the *spatiotemporal* interactions being *direct manipulations*. Among purely *spatial* interactions, while 35% were effectuated by *direct manipulations*, over 54% utilized the indirect *demonstrative* actions such as manipulating motion trails and emission cones. While non-gestural *widget use* was requested for nearly 45% of the *temporal* interactions, participants still used *demonstrative* gestures such as stretching timelines for 40% of those. For *abstract* interactions, *widget uses* jumped to 50%, while 25% of the actions were *semaphoric*.

Gesture *forms* with a *static pose* were generally utilized for *interface*-related tasks only (10/11 gestures). *Static pose and path* gestures were utilized for diverse tasks ranging from *direct manipulations* to pose and perform, to *demonstrative* interactions specifying the rate of smoke emission, and to *semaphoric* actions for duplication. A *dynamic pose and path* was used for physics-affecting *direct manipulations* such as throwing and shattering, for high-bandwidth gestures defining multiple *abstract* properties with a single interaction, and for dragging and dropping *interface* elements.

Bimanual gestures tended to involve a *static pose and path*, (over 90%). Among gestures with *symmetric* hand motion—which formed 64% of all *bimanual* gestures—*spatial* effects such as scaling were the most common (57%), followed by *spatiotemporal* ones (25%) such as performing a rotation (Fig. 4c).

A unique aspect of our experiment compared with existing gesture studies is the lack of predefined operations that the gestures are meant to execute. To help us identify relationships between our taxonomies (defined above) and key tasks in animation authoring processes, we identified 39 unique *atomic operations* from the elicited interactions. Table 3 describes the distribution of various taxomonic classes across these operations. Detailed descriptions of the operations and the process we used to arrive at them are given in the supplemental text.

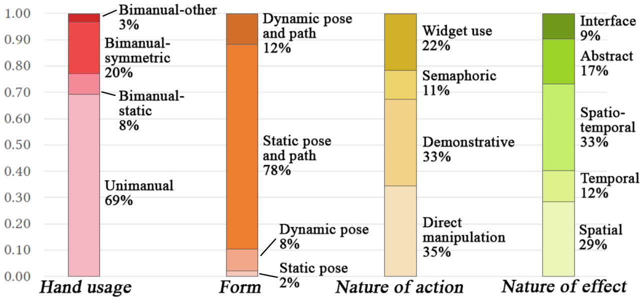


Figure 6. Distribution of gestures and all interactions in the taxonomies.

Salient Observations and Discussion

The freedom afforded to participants and the think-aloud protocol allowed informal observation of high-level trends.

Expressiveness of gestures

We saw significant variation between the number of DoFs controlled by a single gesture. While many gestures specified a single parameter such as particle emission speed, we also observed high-bandwidth gestures manipulating many spatiotemporal parameters simultaneously, such as describing the emission curve, speed, spread and randomness using a single gesture (Figure 1a–b). In the middle, for example, gestures described motion paths in space ($\mathbb{R}^3$) or space-time ($\mathbb{R}^4$), and 3D rotations (4 DoFs: axis and speed).

Expectation and control of physics

Almost all participants had an expectation of interactive physics simulations. In particular, users expected gravity, deformation and contact modelling, and they presumed their ability to directly apply linear and rotational impulses to objects. Participants also attempted to set physical and simulation parameters that would then modify the resulting simulated behaviours. For example, by changing the physical properties, e.g., mass or stiffness of the bouncing ball, or by changing simulation parameters, e.g., "glue strength" of the tearing edges in the sheet animations. Some gestures demonstrated the desired physical behaviour by both direct action and by providing key examples. For instance, P12 described the motion of the pieces in the shattering sheet animation by using their hands as claws to break the surface and then moving them along the desired path of the pieces to provide example motion. Finally, another common class of physics-based gestures directly demonstrated the desired motion by providing examples; for instance by moving one bubble in space and then expecting this to induce the degree of randomness for all the emitted bubbles (Fig. 4j).

Use of traditional widgets

Animators frequently wanted to use traditional widgets to manipulate abstract and temporal parameters. While this can partly be attributed to habit or the difficulty of coming up with gestures for abstract features, animators also note that sliders and numeric inputs allow finer parameter control. P7, P9 also expressed the need for standard terminology for communicating with peers and interoperability with desktop software.

Diegetic interface elements

In immersive interfaces, the term *diegetic* describes an interface which exists "where the action is" [42]. A number of *demonstrative* interactions involved imaginary high-dimensional diegetic interface elements that animators manipulated gesturally. Typical examples include motion trails, onion skins, emitter-attached UI, force fields, and standard anchors for 3D manipulation. Similar high-DoF diegetic widgets have been described, for example, by Conner et al. [11] for manipulating 3D objects and by Sheng et al. [44] for a sculpting interface utilizing physical props. Further, animators frequently desired traditional widgets to be *diegetically positioned*. For example, showing a *properties* menu directly above a selected object.

Gesture overlap

We observed some overlap between gestures elicited from different participants, as well as between different operations performed by the same participant. That is, participants would specify the same (or similar) gestures for different effects. Some of these ambiguities could potentially be resolved with carefully designed diegetic UIs. For example, P3 used a common "point at emitter and move finger up/down" gesture to specify emission density as well as frequency. Other overlaps could be reduced by contextual mode-switching, such as when P1 and P6 used a "double-tap" gesture for selection and bringing up a diegetic panel, respectively.

	Hand Usage					Form					Nature of Action				Nature of Effect				
	U	B.St	B.Sy	B.O	N/A	S	S.P	D	D.P	N/A	DM	Dt	Sm	W	Sp	T	S.T	A	I
Abstract parameter manip.	4	0	0	0	0	0	4	0	0	0	1	1	2	0	0	0	1	3	0
Attaching particle to emitter	3	1	0	1	1	0	4	0	1	1	5	0	0	1	0	0	1	5	0
Bringing up a menu	1	0	1	0	1	1	1	0	0	1	0	0	2	1	0	0	0	0	3
Characterizing object as emitter	4	0	0	0	1	0	1	2	1	1	0	1	3	1	0	0	0	5	0
Coupling	7	1	1	1	4	0	2	5	3	4	0	6	3	5	1	0	1	11	1
Decoupling	4	0	1	0	5	0	5	0	0	5	1	0	4	5	1	0	0	9	0
Duplication	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0
Emission curve desc.	19	4	5	0	0	0	22	1	5	0	6	22	0	0	17	0	10	1	0
Emission density desc.	11	1	1	1	12	1	11	1	1	12	4	5	5	12	2	0	2	20	2
Emission frequency desc.	8	4	2	2	7	0	10	2	4	7	2	10	4	7	1	17	0	4	1
Emission lifetime spec.	5	2	0	0	7	0	6	0	1	7	0	7	0	7	0	0	2	12	0
Emission speed desc.	15	2	5	0	8	0	17	0	5	8	3	17	2	8	0	19	10	1	0
Emission spread desc.	22	0	20	0	2	0	36	1	5	2	4	37	1	2	36	0	6	2	0
Group path desc.	6	0	3	0	3	0	6	1	2	3	2	7	0	3	3	0	8	1	0
Group shape manip.	2	0	0	0	3	0	1	1	0	3	1	1	0	3	1	0	4	0	0
Grouping	4	0	0	3	0	0	7	0	0	0	0	7	0	0	0	0	0	1	6
Local region selection	4	0	0	2	0	0	4	0	2	0	3	2	1	0	1	0	2	1	2
Local shape manip.	7	0	2	2	1	0	10	0	1	1	10	1	0	1	0	0	10	2	0
Motion path desc.	73	0	3	1	11	0	65	3	9	11	60	11	6	11	18	5	54	8	3
Motion path manip.	6	0	1	0	0	0	6	1	0	0	3	4	0	0	5	0	2	0	0
Moving a scaling anchor	1	1	0	0	0	0	2	0	0	0	1	1	0	0	1	0	1	0	0
Object rotation	43	15	13	0	7	0	53	0	18	7	61	8	2	7	10	2	64	2	0
Object scaling	0	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0
Object selection	17	0	0	0	0	7	2	7	1	0	3	2	12	0	0	0	1	0	16
Object shape manip.	22	13	30	1	8	0	59	5	2	8	49	13	4	8	40	0	23	7	4
Pause/Play	5	0	1	0	0	1	3	2	0	0	0	0	6	0	0	0	0	0	6
Posing (move w/o record)	0	0	2	0	0	0	2	0	0	0	2	0	0	0	2	0	0	0	0
Randomness desc.	26	1	1	2	20	0	18	3	9	20	7	21	1	21	5	0	35	7	3
Record ON/OFF toggling	6	0	0	0	5	1	4	0	1	5	3	1	2	5	3	0	0	2	6
Scrubbing the timeline	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1
Secondary emission desc.	5	0	0	0	5	0	5	0	0	5	4	1	0	5	1	0	7	2	0
Timing manip.	11	1	3	0	3	0	15	0	0	3	3	11	1	3	0	13	5	0	0
Topological change spatial desc.	28	1	4	0	2	0	30	0	3	2	7	25	1	2	28	0	4	2	1
Topological change timing desc.	4	0	3	0	16	0	5	0	2	16	3	3	1	16	2	12	5	4	0
UI panel manip.	3	0	0	0	0	0	0	1	2	0	0	3	0	0	0	0	0	0	3
Vibrational amplitude spec.	9	2	3	0	2	0	10	3	1	2	10	2	2	2	7	2	6	1	0
Vibrational frequency spec.	6	2	0	1	6	0	7	2	0	6	7	0	2	6	1	5	8	1	0
World rotation	0	0	2	0	0	0	2	0	0	0	1	1	0	0	0	0	1	0	1
World scaling	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1

Table 3. Observed atomic operations with classification of the interactions utilized for each. The numbers in the cells represent the observed frequencies of each taxonomic class for the actions performed by our participants. U–*unimanual*, B.St–*bimanual-static*, B.Sy–*bimanual-symmetric*, B.O–*bimanual-other*, N/A–*no gesture*, S–*static pose*, S.P–*static pose and path*, D–*dynamic pose*, D.P–*dynamic pose and path*, DM–*direct manipulation*, Dt–*demonstrative*, Sm–*semaphoric*, W–*widget interaction*, Sp–*spatial*, T–*temporal*, S.T–*spatiotemporal*, A–*abstract*, I–*interface*.

Comparison with Existing Studies on Gestures

We noticed revealing similarities between our findings for mid-air gestures and those of earlier studies on touch and motion gestures. Wobbrock et al. [60] reported that users rarely cared about the number of fingers used for touch gestures, instead relying on the *form* of the gesture. Our findings for mid-air gestures reinforce theirs. For instance, when directly manipulating an object, participants' hand pose either mimicked grabbing the object physically or was a canonical pose such as pointing with the index finger. A similar effect was noticed for some gestures with dynamic hand poses—participants would, for example, flutter their fingers to add noise to smoke, but only the quantity of fluttering was deemed important and not the exact spatial relationship between the fingers. Our findings on gesture expressiveness follow Brouet et al.'s [8]. While participants did use gestures rich in spatiotemporal information, the number of DoFs in the intended effect was never close to the theoretical limit of 27 (54 for bimanual). Our bimanual gestures also fit neatly into the *bimanual-static* or *bimanual-symmetric* categories they defined for multi-touch. We also noticed that semaphoric and demonstrative gestures for logically opposite tasks tended to be geometrically mirrored—move hand left/right to scrub forward/back in time, move hand up/down to increase/decrease the value of a numeric parameter, bring the thumb and index finger together/move away to couple/decouple objects, etc. Similar findings have been reported for motion gestures [41] and touch gestures [60].

Comparisons can also be drawn with studies on mid-air gestures targeting other applications. Aigner et al. [3] elicited a number of pointing gestures for human-to-human interaction. In contrast, our participants generally chose to directly grab objects. Vatavu [56] suggests augmenting mid-air gestures for TV control with other modalities such as on-screen widgets or a traditional remote control. Our findings also suggested that a *gesture-dominated* interface prudently utilizing traditional widgets and spatial UI can be more useful than a purely gestural interface. Lastly, owing to the ubiquity of touch devices, some gestures seemed to be inspired by touch gestures. For example, participants air-tapped for selection and "pinch to zoom" to scaling. A similar observation was made by Troiano et al. [53] for gestures manipulating elastic displays.

DESIGN GUIDELINES FOR GESTURAL ANIMATION

We consolidate our formal and informal observations into a set of design guidelines for VR animation interfaces that employ mid-air gestures. We hope these guidelines serve as a useful starting point for future experiments and the design of new immersive animation systems.

Breakdown of Interactions for Various Effects

Our study suggests direct manipulation as the primary interaction mode for spatiotemporal operations (Fig. 4a,d). Such interactions allow users to directly perform complex motions in spacetime. For operations that are either spatial or temporal (but not both), a combination of widget use, direct manipulation, and abstract demonstrative gestures may be appropriate. In addition, editable diegetic representations of spatiotemporal properties (e.g., editable motion trails) are effective for visualizing and refining previously-authored motions.

Contextual Interaction Bandwidth

Most animators in our study adopted a coarse-to-fine workflow where they quickly specified a rough scene before delving into the details. To support this process, VR animation systems should provide high-bandwidth, direct manipulation gestures to quickly specify the overall spatiotemporal properties of an animation, and then allow layering of other, often finer, details with gestures that are scoped to control just the relevant, local, spatial and/or temporal properties. Moreover, context-dependent gestures can help reduce friction in such workflows. For example, beginning with a point-and-move gesture to perform the rough spatiotemporal path of an object, the same gesture can then be re-used to perform fine-grain control of the spatial positioning of the motion trail or small adjustments to the object's motion specified in its local coordinate frame. In each context, the system should infer which attributes of the gesture to use and ignore; e.g., when editing the motion trail, the execution speed of the gesture is not relevant.

Natural Interactions with Physics

Physical systems in VR animation utilize a wide range of physical parameters to obtain expressive simulations. However, our study indicates that animators generally do not want to consciously manipulate all such physical variables. Instead we see animators focusing individually on only particular, generally high-level, aspects of physical simulation output, with the expectation that the underlying simulation system will automatically make reasonable choices by "taking care" of the rest of the parameters. Moreover, we observed many different types of interactions related to physics, from direct (e.g., throwing an object as in Figure 4a) to indirect (e.g., changing noise, changing emission frequency as in Figure 4f). This suggests that VR animation systems should support real-world physical behaviors with reasonable default settings and that users should be able to interact with simulations in various ways while seeing real-time results. At the same time, it is important to provide expressive controls that allow animators to art-direct and exaggerate physics-based effects beyond the boundaries of realistic defaults when necessary.

Hand Pose Granularity and Gesture Recognition

A practical observation from our study is that most of the gestures did not encode information in the nuances of hand shape. Participants were largely ambivalent towards the hand poses they used for a large number of gestures. Further, most of the observed gestures utilized only a few important poses (Figure 7), that can be easily distinguished.Thus, we observe that users are both forgiving of high-frequency noise in pose recognition and thus possibly cheaper, unsophisticated, hardware and coarse recognition can potentially serve as a good starting point for gestural interfaces.

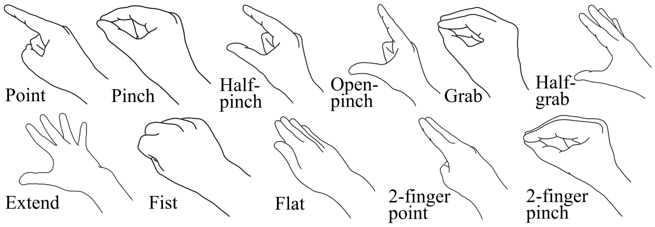


Figure 7. Typical hand poses observed in our study.

We also note that participants were willing to learn a few key poses to disambiguate between commonly used commands such as manipulating pose vs. performance. In turn this suggests flexibility in the choice of poses assigned to operations, which can be exploited to maximize recognition accuracy.

Controller-based Interfaces

Finally, while our experiments were conducted to understand bare-handed gestures, some findings can potentially benefit controller-based interaction as well. Prior research has shown that grip can be leveraged as an implicit dimension conveying some user intention as they interact with physical devices [48, 59, 46]. More recently, multi-purpose haptic rendering [10] controllers demonstrate how hand poses can be utilized to define gestures similar to bare-handed gestures. Beyond static postures, future controller designs can also potentially leverage dynamic poses and high-bandwidth gestures to simulate the expressiveness of bare-handed input, while retaining the benefits of haptic controllers. For example, participants defined the amount of smoke dispersion by interpolating between the "pinch" and "extend" poses (Fig. 7). Controllers detecting a continuous pose change via a proxy measure can enjoy an input effectively as rich as bare-hands for such interactions. Given current limitations in hand-gesture tracking technologies [67], we explore this guideline further in the next section.

MAGICALHANDS: DESIGN AND IMPLEMENTATION

To investigate our design guidelines, we developed a prototype VR animation system—MagicalHands. We use MagicalHands to explore concrete instantiations of our guidelines and to assess the feasibility and usability of a subset of the novel gestures we observed. In this proof-of-concept tool, we focus on the creation and manipulation of particle systems. Based on our formative study, we implemented eleven interactions, including eight gestures (from direct manipulation to abstract demonstrations) and three abstract operations utilizing 3D UI.

Interaction Design Concepts

We first describe the UI concepts utilized by our system.

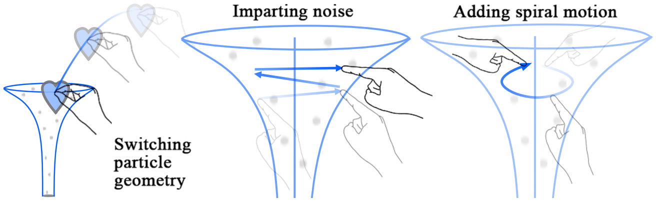


Figure 8. Implemented gestures for particle system manipulation.

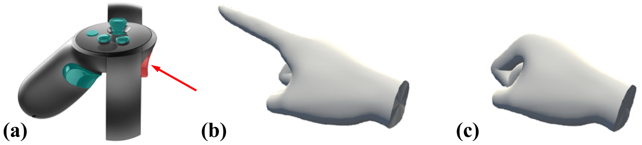


Figure 9. Touch-sensitive buttons and triggers on the Oculus Touch controller can be mapped to hand poses. For example, the absence of touch on the front trigger (a, red arrow) is interpreted as the index finger being extended (b), while its presence is interpreted as the finger being bent (c).

Visual Entities. In an immersive setting like ours, we use several types of visual entities, including scene objects (e.g., bubbles), animation primitives (e.g., particle emitter), and UI elements (e.g., playback control buttons). As suggested by the design guideline on breakdown of interactions, explicit diegetic realizations of hand gestures and performances could also be used as visual entities for editing, though we did not include them in our prototype. Access to objects for animation is provided via an *object shelf* positioned at a fixed location in the immersive world. The shelf contains animation-ready 3D meshes imported into the system and special *emitter* objects which can be used to spawn particle systems.

Extrinsic and Intrinsic Attributes. To avoid visual clutter, *extrinsic* entities (e.g., 3D models, shelf, UI controls) have persistent visual affordances, and can be directly accessed using gestures for common animation operations. *Intrinsic* entities (e.g., emission curve, spread, and noise) are made visible and operable in context by gesturally selecting the pertinent object first. Direct manipulation of an extrinsic attribute in MagicalHands is naturally invoked by gesturally grabbing its visible affordance. A selection gestures can be used to select objects and enable intrinsic attribute manipulation.

Creation Process Freedom. Animation of object attributes in an interactive setting is typically authored using keyframes (for smooth low frequency changes); human performance (for high frequency detail and timing); and simulation (for high-DoF changes governed by the laws of physics). Our implementation supports all three, driven off an animation control widget with familiar UI and gesturally-activated extrinsic play/pause/record/scrub attributes.

Eulerian and Lagrangian Gestures. In simulation, *Lagrangian* specification directly defines an object's attribute in time, while *Eulerian* specification describes the properties of the ambient space that an object exists in. In MagicalHands, the rigid motion of an object is specified in the former manner by grabbing it and performing/keyframing its attributes. We additionally advocate animating position and orientation separately, as cognitive and anatomic limitations make it difficult to perform arbitrarily complex motion trajectories. Gestures impacting *intrinsic* attributes such as emission properties impact particles in an *Eulerian* manner, influencing the space around an emitter rather than directly manipulating the particles. The supplementary material provides more details.

User Interaction

We now describe how users interact with the above concepts.

Instantiation and Deletion. Dragging an object from the shelf—by pointing at it, pinching it, or grabbing it—and then dropping it into the scene instantiates a copy of that object. Deletion involves performing this gesture in reverse.

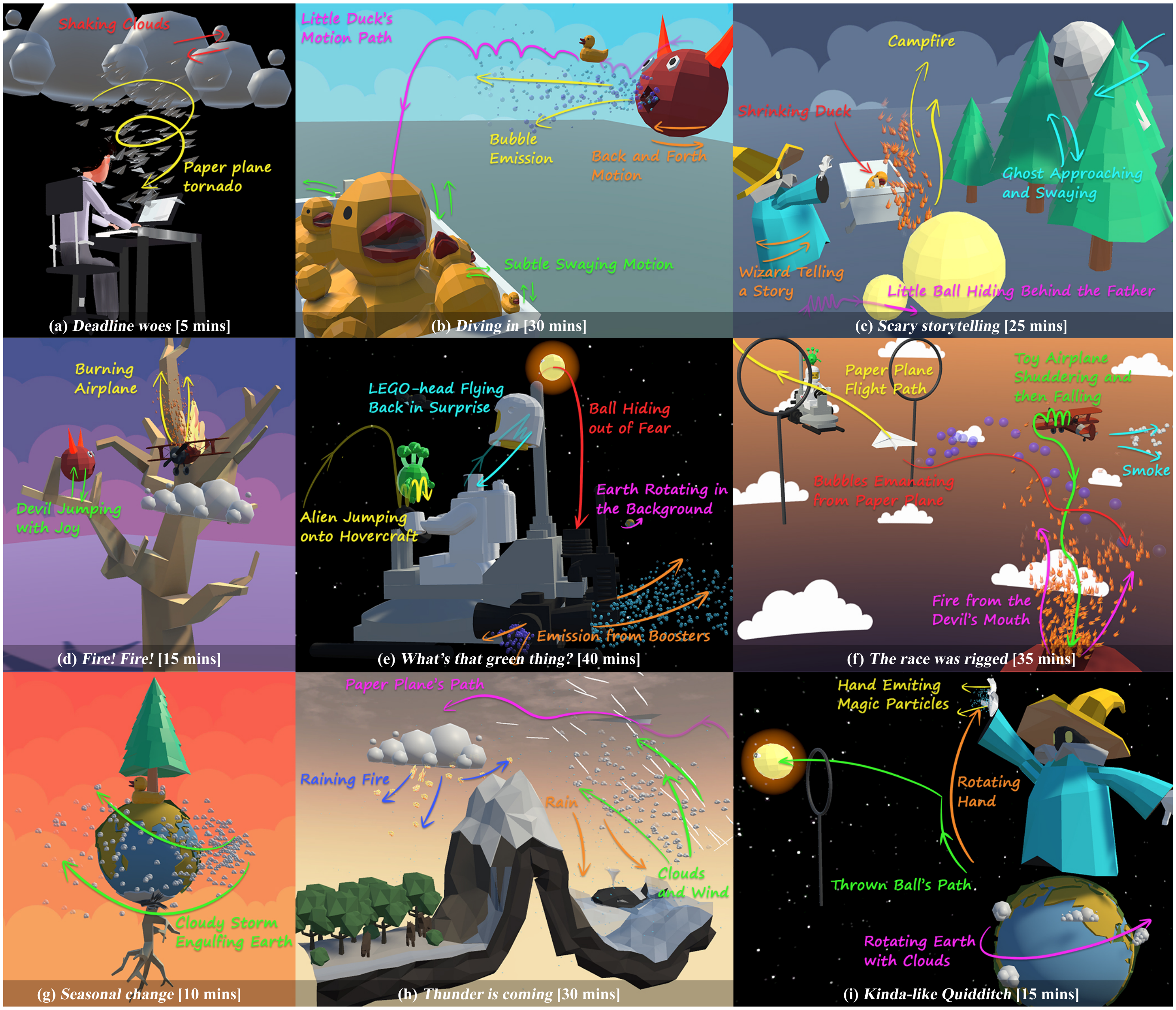
Selection. Users can *tap* objects (Fig. 4h) in the scene to select. For particle systems, this visualizes the current emission curve and spread, and enables emitter-specific interactions.

Direct Manipulation for Natural Interactions. We implemented three direct manipulation performance gestures to record a motion. Based on our design guidelines, we support direct performance of a motion path (Fig. 4a). The user pinches onto an object with the dominant hand and then moves it to **translate** the object over time. Pinching with both hands and moving the hands closer together or further apart (Fig. 4b[1]) **scales** the object uniformly. Finally, using the dominant hand in a pointing pose defines an axis of rotation and the motion of the non-dominant hand around this axis (Fig. 4c[1]) **rotates** the target object. Recall that our study suggests implicit selection for manipulating such *extrinsic* properties, and thus, an explicit selection gesture is not required for direct manipulation.

High-Bandwidth & Demonstrative Gestures for Particles. Instantiating an emitter object creates a particle system. As indicated by our design guideline on physical simulations, this interaction automatically produces an animated result with default particle geometry, and default parameters for emission curve, speed, and spread. In addition, the simulation provides real-time feedback and allows users to modify it with various interactions. For example, they can move the emitter itself and drag and drop objects from the shelf into the emitter to **attach particle to emitter** (Fig. 8). Selecting an emitter visualizes the current emission curve and volume, and enables specialized, context-dependent gestures.

— A *high-bandwidth gesture* describes the **emission curve, speed, and spread** intrinsics in a single action. Users position both hands close to an emitter in fist pose and move away to perform the gesture (Fig. 4d[1]). The curve swept out by the mid-point of the imaginary line joining the hands defines the emission curve and speed, while the distance between them defines the emission spread along the curve.

— Users can perform an abstract demonstration by moving their hand inside the emission volume in a pinch, point, or fist shape to **impart noise** (Fig. 8). Performing a **spiral motion** makes the particles swirl around the emission curve. Moving the hand along a single dominant axis activates the noise gesture, while moving in a spiral activates the spiral force gesture. This inference is automatic.


Shaking Clouds
Paper plane tornado
Little Duck's Motion Path
Bubble Emission
Back and Forth Motion
Subtle Swaying Motion
Campfire
Shrinking Duck
Ghost Approaching and Swaying
Wizard Telling a Story
Little Ball Hiding Behind the Father
Burning Airplane
Devil Jumping with Joy
LEGO-head Flying Back in Surprise
Ball Hiding out of Fear
Alien Jumping onto Hovercraft
Earth Rotating in the Background
Emission from Boosters
Paper Plane Flight Path
Toy Airplane Shuddering and then Falling
Smoke
Bubbles Emanating from Paper Plane
Fire from the Devil's Mouth
Cloudy Storm Engulfing Earth
Paper Plane's Path
Raining Fire
Rain
Clouds and Wind
Hand Emiting Magic Particles
Rotating Hand
Thrown Ball's Path
Rotating Earth with Clouds

(a) *Deadline woes* [5 mins]

(b) *Diving in* [30 mins]

(c) *Scary storytelling* [25 mins]

(d) *Fire! Fire!* [15 mins]

(e) *What's that green thing?* [40 mins]

(f) *The race was rigged* [35 mins]

(g) *Seasonal change* [10 mins]

(h) *Thunder is coming* [30 mins]

(i) *Kinda-like Quidditch* [15 mins]

Figure 10. Stills from animations authored using MagicalHands, along with execution time (asset arrangement + animating + experimentation) for each clip. Author creations (a, b, e, i), and participant creations (c, d, f, g, h). Please see the supplementary video for the animated scenes. Man typing on computer (a) model ©Chuantao Xu; used with permission.

Egocentric Interface for Navigation. We position a UI panel close to the non-dominant hand of the user, with buttons for **playing or pausing** the animation clip, and another for **toggling record ON or OFF**. When recording is ON, the direct manipulation gestures are treated as spatiotemporal *performances* and the speed of the gesture is transferred onto the target object. In contrast, when recording is OFF, these gestures serve as *posing* tools—defining a keyframe at the current frame. The panel also contains timeline scrubbing buttons.

Setup and Implementation Details

Implementation of the gestures requires robust, real-time 3D hand recognition capabilities. Fortunately, our guidelines suggest that most gestures utilize only a few important poses. Thus, mindful of current hand recognition technology limitations, we use the hand-held Oculus Touch controllers. They provide larger tracking volume (compared to the Leap Motion, for example) and reliable hand-pose information which allows us to test our guidelines in this setup—sacrificing some gesture recognition capability for robustness. Note that gestures which require continuous pose tracking and/or unconventional hand poses are rendered infeasible by this setup. This reduces the possible set of gestures we currently implement in the system.

Our implementation utilizes a layer of abstraction over the raw tracking data so that the inference system only uses hand pose information from the controller (Fig. 9). Practically, this means that our system can be easily adapted to run with hand-pose data generated by bare-handed tracking techniques, motion-captured gloves, or other lightweight sensors.

We use the position of the fingertips w.r.t the wrist and the distance between them to infer hand poses among point, pinch, fist, and extended (neutral) poses (Fig. 7). 6-DoF hand tracking data combined with this inferred pose is utilized for gesture recognition. For direct manipulation gestures, transformations are applied to the objects in their local space, which allows chaining together different objects by parenting. In the current prototype, parental hierarchy has to be manually defined while importing objects. Animations are played back to the users in a loop, and the duration of the clip is simply determined by the performed gestures: the clip initially consists of a single frame, and performing manipulations whose length goes past the clip duration automatically adds frames to the clip to record the entire gesture. The translation, rotation, and scaling gestures define a dense set of keyframes when used as performance (record ON), and a sparse set when used to pose (record OFF). The system was implemented in C# using Unity.

Testing and Results

We invited four artists to try our MagicalHands system in an informal setting. P1 had formal training in animation, P2–3 had amateur experience with desktop-based animation tools, and P4 was a professional industrial designer, with 5 years of experience with VR-based 3D design and modelling. However, none had experience using VR for animation tasks.

Users were given a short tutorial explaining how they can utilize hand gestures for various tasks. They could then freely explore the system and create animations. Users had access to 35 static objects in the shelf to use in their creations, including three particle emitters with different emission densities. Figures 1 and 10 show stills from the clips created by the participants, as well as those created by the authors of this work. Depending on scene complexity, creating these clips took between 5 and 40 minutes, excluding planning time. The supplemental material includes animation videos, as well as the step-by-step creation process for two author-created scenes.

User Feedback

In general, users really liked the the use of VR and live performance aspects of the system. P2 found that *“just placing objects [in VR] is already interesting...being able to animate directly is really cool...”*. The ability to directly control particle systems was also appreciated. All the users found gestural manipulation of particle systems useful. P2 found it *“extremely fun to work with”*, while P1, who had experience with Maya [5], found the *“gestures in VR make a lot of sense since drawing 3D splines on a 2D screen is really hard”*. More generally, they thought that *“drawing the space curve live was useful for timing the particle emission and [for] translations”*.

P4 had insights about interacting with higher-level abstractions of gestures: *“I need to see my gesture materialize...it should be like a sculpture...”*. While other gestures were appreciated, users found the rotation gesture a bit mentally taxing. P3 commented that *“being able to define a particular axis [of rotation] is useful”* but alternatives such as *“being able to use one hand as a pivot point and other to [implicitly] define the axis”* could provide more intuitive control. Still, users successfully utilized the gesture, and their creations incorporated both performed (P1) and keyframed (P3) rotations.

CONCLUSION AND FUTURE WORK

Our work expands the body of HCI research on hand gestures to animation authoring. From our elicitation study, we have identified and categorized a broad spectrum of gestural interactions that can be used to depict complex dynamic effects in VR. In addition to gestures, our observations suggest that traditional widgets and spatial UI elements are useful for fine control over specific parameters. Based on these findings, we derived a set of design guidelines to inform the design of gesture-based VR animation authoring tools, and developed a prototype system to test our guidelines and observed gestures. Our user study demonstrates the effectiveness and potential of hand gestures to author animation in VR.

Looking forward, our work points at several interesting directions for future research. While our study describes a range of potential gestures for animation, it would be valuable to conduct additional experimental work on the usability of such gestures. Another interesting research direction is the evaluation of mid-air gesture taxonomies, including ours, to identify gaps and compare different classification strategies, similar to the recent work of Kim et al. [26] on shape-changing interfaces. In parallel, we hope our design guidelines inspire system builders to develop novel gesture-based VR animation tools; and that our MagicalHands system can help in these future efforts. Such efforts would likely require new gesture tracking and recognition algorithms and novel user interfaces that combine mid-air gestures with diegetic and traditional widgets. In this vein, one specific sub-area to investigate is gestural control of physical simulations. Here, a key challenge is co-developing robust and controllable simulation methods that predictably respond to gestures. In order to facilitate further development of such experimental work, we have released our source code at `https://github.com/rarora7777/MagicalHands`.

Finally, although our work focuses on animating physical phenomena, some of our findings may apply more broadly to other dynamic effects, such as lighting changes and mechanical movements. Moreover, some gestures could translate to non-VR animation tasks provided that users have effective ways to target gestures to desired objects. Validating these ideas is an exciting future work direction that would further expand the scope of gestural interactions for content creation.

ACKNOWLEDGEMENTS

We thank Seth Walker for illuminating discussions which helped shaped our prototype, our study participants for their patience, and the anonymous reviewers for helping improve this paper. We are also immensely thankful to these amazing artists for their CC-BY artwork: Taylor Wilson (*Alien*), Fernando Dominguez (*Wizard*), Drifting Nitro (*Drone*), Andrew Julian (*Ghost*), Ben Mitchell (*Lego astronaut*), Danny Bittman (*Pine tree*), Anonymous (*Purple paper plane*), Junior IT (*Sad ball*), Jerard Bitner (*Bathtub*), and Google (*Campfire*, *Dead tree*, *Earth*, *Rubber duck* and *Toy airplane*). Rahul Arora was partially supported by an Adobe Research Fellowship.

REFERENCES

[1] Adobe. 2018. Character Animator. https://adobe.com/products/character-animator.html. (2018).

[2] Anne M.R. Agur and Ming J. Lee. 1999. *Grant's Atlas of Anatomy* (10 ed.). Williams and Wilkins.

[3] Roland Aigner, Daniel Wigdor, Hrvoje Benko, Michael Haller, David Lindbauer, Alexandra Ion, Shengdong Zhao, and Jeffrey Tzu Kwan Valino Koh. 2012. *Understanding Mid-Air Hand Gestures: A Study of Human Preferences in Usage of Gesture Types for HCI*. Technical Report MSR-TR-2012-111.

[4] David Akers. 2006. Wizard of Oz for Participatory Design: Inventing a Gestural Interface for 3D Selection of Neural Pathway Estimates. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. ACM, New York, NY, USA, 454–459. DOI:http://dx.doi.org/10.1145/1125451.1125552

[5] Autodesk. 2019. Maya. https://autodesk.com/products/maya/features. (2019).

[6] Yunfei Bai, Danny M. Kaufman, C. Karen Liu, and Jovan Popović. 2016. Artist-directed Dynamics for 2D Animation. *ACM Trans. Graph.* 35, 4, Article 145 (July 2016), 10 pages. DOI: http://dx.doi.org/10.1145/2897824.2925884

[7] Miklós Bergou, Saurabh Mathur, Max Wardetzky, and Eitan Grinspun. 2007. TRACKS: Toward Directable Thin Shells. *ACM Trans. Graph.* 26, 3, Article 50 (July 2007). DOI:http://dx.doi.org/10.1145/1276377.1276439

[8] Rémi Brouet, Renaud Blanch, and Marie-Paule Cani. 2013. Understanding Hand Degrees of Freedom and Natural Gestures for 3D Interaction on Tabletop. In *Human-Computer Interaction – INTERACT 2013*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[9] Jiawen Chen, Shahram Izadi, and Andrew Fitzgibbon. 2012. KinÊtre: Animating the World with the Human Body. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 435–444. DOI: http://dx.doi.org/10.1145/2380116.2380171

[10] Inrak Choi, Eyal Ofek, Hrvoje Benko, Mike Sinclair, and Christian Holz. 2018. CLAW: A Multifunctional Handheld Haptic Controller for Grasping, Touching, and Triggering in Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 654.

[11] Brookshire D. Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, and Andries van Dam. 1992. Three-dimensional Widgets. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics (I3D '92)*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/147156.147199

[12] Richard C Davis, Brien Colwell, and James A Landay. 2008. K-sketch: a'kinetic'sketch pad for novice animators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 413–422.

[13] Mira Dontcheva, Gary Yngve, and Zoran Popović. 2003. Layered Acting for Character Animation. *ACM Trans. Graph.* 22, 3 (July 2003), 409–416. DOI: http://dx.doi.org/10.1145/882262.882285

[14] Facebook. 2019. Quill. https://quill.fb.com. (2019).

[15] Joseph Gilland. 2009. *Elemental Magic, Volume I: The Art of Special Effects Animation*. Routledge.

[16] Robert Held, Ankit Gupta, Brian Curless, and Maneesh Agrawala. 2012. 3D Puppetry: A Kinect-based Interface for 3D Animation. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA. DOI:http://dx.doi.org/10.1145/2380116.2380170

[17] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. 1994. A Survey of Design Issues in Spatial Input. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology (UIST '94)*. ACM, New York, NY, USA, 213–222. DOI: http://dx.doi.org/10.1145/192426.192501

[18] Christian Holz and Andrew Wilson. 2011. Data miming: inferring spatial object descriptions from human gesture. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 811–820.

[19] Takeo Igarashi, Tomer Moscovich, and John F Hughes. 2005. As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)* 24, 3 (2005).

[20] Søren Qvist Jensen, Andreas Fender, and Jörg Müller. 2018. Inpher: Inferring Physical Properties of Virtual Objects from Mid-Air Interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/3173574.3174104

[21] Maria Karam and m.c. schraefel. 2005. *A taxonomy of gestures in human computer interactions*. Technical Report ECSTR-IAM05-009.

[22] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George Fitzmaurice. 2014a. Kitty: Sketching Dynamic and Interactive Illustrations. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/2642918.2647375

[23] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, Shengdong Zhao, and George Fitzmaurice. 2014b. Draco: Bringing Life to Illustrations with Kinetic Textures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 351–360. DOI: http://dx.doi.org/10.1145/2556288.2556987

[24] Rubaiat Habib Kazi, Kien Chuan Chua, Shengdong Zhao, Richard Davis, and Kok-Lim Low. 2011. SandCanvas: a multi-touch art medium inspired by sand animation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM.

[25] Micky Kelager. 2006. *Lagrangian fluid dynamics using smoothed particle hydrodynamics*. Master's thesis. University of Copenhagen: Dept. of Computer Science.

[26] Hyunyoung Kim, Celine Coutrix, and Anne Roudaut. 2018b. Morphees+: Studying Everyday Reconfigurable Objects for the Design and Taxonomy of Reconfigurable UIs. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 619, 14 pages. DOI: http://dx.doi.org/10.1145/3173574.3174193

[27] Yongkwan Kim, Sang-Gyun An, Joon Hyub Lee, and Seok-Hyung Bae. 2018a. Agile 3D Sketching with Air Scaffolding. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 238.

[28] Hoo Yong Leng, Noris Mohd Norowi, and Azrul Hazri Jantan. 2017. A User-Defined Gesture Set for Music Interaction in Immersive Virtual Environment. In *Proceedings of the 3rd International Conference on Human-Computer Interaction and User Experience in Indonesia (CHIuXiD '17)*. ACM, New York, NY, USA. DOI:http://dx.doi.org/10.1145/3077343.3077348

[29] Noah Lockwood and Karan Singh. 2012. Finger Walking: Motion Editing with Contact-based Hand Performance. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '12)*. Eurographics Association, Germany. http://dl.acm.org/citation.cfm?id=2422364

[30] JR Martin, VM Zatsiorsky, and ML Latash. 2011. Multi-finger interaction during involuntary and voluntary single finger force changes. *Experimental brain research* 208, 3 (2011), 423–435.

[31] David McNeill. 1992. *Hand and mind: What gestures reveal about thought*. University of Chicago press.

[32] Ali Momeni and Zachary Rispoli. 2016. Dranimate: Paper Becomes Tablet, Drawing Becomes Animation. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 3735–3737. DOI:http://dx.doi.org/10.1145/2851581.2890267

[33] Nvrmind. 2019. AnimVR. https://nvrmind.io. (2019).

[34] Ollie. 2019. Ollie VR. https://ollievr.com. (2019).

[35] Zherong Pan, Jin Huang, Yiying Tong, Changxi Zheng, and Hujun Bao. 2013. Interactive Localized Liquid Motion Editing. *ACM Trans. Graph.* 32, 6 (Nov. 2013). DOI:http://dx.doi.org/10.1145/2508363.2508429

[36] R. Parent. 2012. *Computer Animation: Algorithms and Techniques*. Elsevier Science. https://books.google.ca/books?id=DudZtbOD2gMC

[37] Frederic Pighin and J.P. Lewis. 2006. Performance-Driven Facial Animation. In *SIGGRAPH Courses*. ACM.

[38] Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. 2013. User-Defined Gestures for Augmented Reality. In *Human-Computer Interaction – INTERACT 2013*, Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 282–299.

[39] Jun Rekimoto. 2002. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 113–120. DOI: http://dx.doi.org/10.1145/503376.503397

[40] Isabel Benavente Rodriguez and Nicolai Marquardt. 2017. Gesture Elicitation Study on How to Opt-in & Opt-out from Interactions with Public Displays. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17)*. ACM, New York, NY, USA, 32–41. DOI: http://dx.doi.org/10.1145/3132272.3134118

[41] Jaime Ruiz, Yang Li, and Edward Lank. 2011. User-defined Motion Gestures for Mobile Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 197–206. DOI: http://dx.doi.org/10.1145/1978942.1978971

[42] Paola Salomoni, Catia Prandi, Marco Roccetti, Lorenzo Casanova, Luca Marchetti, and Gustavo Marfia. 2017. Diegetic user interfaces for virtual environments with HMDs: a user experience study with oculus rift. *Journal on Multimodal User Interfaces* 11, 2 (2017), 173–184. DOI:http://dx.doi.org/10.1007/s12193-016-0236-5

[43] Steven Schkolne, Michael Pruett, and Peter Schröder. 2001. Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM, New York, NY, USA, 261–268. DOI: http://dx.doi.org/10.1145/365024.365114

[44] Jia Sheng, Ravin Balakrishnan, and Karan Singh. 2006. An Interface for Virtual 3D Sculpting via Physical Proxy. In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia (GRAPHITE '06)*. ACM, New York, NY, USA, 213–220. DOI: http://dx.doi.org/10.1145/1174429.1174467

[45] SideFX. 2019. Houdini. https://sidefx.com/products/houdini. (2019).

[46] Hyunyoung Song, Hrvoje Benko, Francois Guimbretiere, Shahram Izadi, Xiang Cao, and Ken Hinckley. 2011. Grips and gestures on a multi-touch pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1323–1332.

[47] Christian Steins, Sean Gustafson, Christian Holz, and Patrick Baudisch. 2013. Imaginary devices: gesture-based interaction mimicking traditional input devices. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*. ACM, 123–126.

[48] Brandon T Taylor and V Michael Bove Jr. 2009. Graspables: grasp-recognition as a user interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 917–926.

[49] The Blender Foundation. 2019. blender. `https://blender.org/features/simulation`. (2019).

[50] S Thieffry. 1981. Hand gestures. *The Hand (R. Tubiana, ed.)* 488 (1981).

[51] Thinksquirrel. 2019. Fluvio. `https://getfluv.io`. (2019).

[52] Matthew Thorne, David Burke, and Michiel van de Panne. 2004. Motion doodles: an interface for sketching character motion. In *ACM Transactions on Graphics (TOG)*, Vol. 23. ACM, 424–431.

[53] Giovanni Maria Troiano, Esben Warming Pedersen, and Kasper Hornbæk. 2014. User-defined Gestures for Elastic, Deformable Displays. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*. ACM, New York, NY, USA, 1–8. DOI:`http://dx.doi.org/10.1145/2598153.2598184`

[54] Tvori. 2019. Tvori VR. `http://tvori.co`. (2019).

[55] Unity Technologies. 2019. Unity. `https://unity.com`. (2019).

[56] Radu-Daniel Vatavu. 2012. User-defined Gestures for Free-hand TV Control. In *Proceedings of the 10th European Conference on Interactive TV and Video (EuroITV '12)*. ACM, New York, NY, USA, 45–48. DOI: `http://dx.doi.org/10.1145/2325616.2325626`

[57] Stephen Voida, Mark Podlaseck, Rick Kjeldsen, and Claudio Pinhanez. 2005. A Study on the Manipulation of 2D Objects in a Projector/Camera-based Augmented Reality Environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 611–620. DOI: `http://dx.doi.org/10.1145/1054972.1055056`

[58] Daniel Wigdor, Hrvoje Benko, John Pella, Jarrod Lombardo, and Sarah Williams. 2011. Rock & Rails: Extending Multi-touch Interactions with Shape Gestures to Enable Precise Spatial Manipulations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1581–1590. DOI: `http://dx.doi.org/10.1145/1978942.1979173`

[59] Raphael Wimmer and Sebastian Boring. 2009. HandSense: discriminating different ways of grasping and holding a tangible user interface. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*. ACM, 359–362.

[60] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. 2009. User-defined Gestures for Surface Computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1083–1092. DOI:`http://dx.doi.org/10.1145/1518701.1518866`

[61] Mike Wu and Ravin Balakrishnan. 2003. Multi-finger and Whole Hand Gestural Interaction Techniques for Multi-user Tabletop Displays. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. ACM, New York, NY, USA, 193–202. DOI: `http://dx.doi.org/10.1145/964696.964718`

[62] Shihong Xia, Lin Gao, Yu-Kun Lai, Mingzhe Yuan, and Jinxiang Chai. 2017. A Survey on Human Performance Capture and Animation. *J. Comput. Sci. Technol.* 32, 3 (2017), 536–554.

[63] Jun Xing, Rubaiat Habib Kazi, Tovi Grossman, Li-Yi Wei, Jos Stam, and George Fitzmaurice. 2016. Energy-Brushes: Interactive Tools for Illustrating Stylized Elemental Dynamics. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 755–766. DOI: `http://dx.doi.org/10.1145/2984511.2984585`

[64] Yukang Yan, Chun Yu, Xiaojuan Ma, Xin Yi, Ke Sun, and Yuanchun Shi. 2018. VirtualGrasp: Leveraging Experience of Interacting with Physical Objects to Facilitate Digital Object Retrieval. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 78.

[65] Xiao Yi, Shengfeng Qin, and Jinsheng Kang. 2009. Generating 3D architectural models based on hand motion and gesture. *Computers in Industry* 60, 9 (2009), 677 – 685. DOI: `http://dx.doi.org/10.1016/j.compind.2009.05.001`

[66] Run Yu and Doug A Bowman. 2018. Force Push: Exploring Expressive Gesture-to-Force Mappings for Remote Object Manipulation in Virtual Reality. *Frontiers in ICT* 5 (2018), 25.

[67] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, Junsong Yuan, Xinghao Chen, Guijin Wang, Fan Yang, Kai Akiyama, Yang Wu, Qingfu Wan, Meysam Madadi, Sergio Escalera, Shile Li, Dongheui Lee, Iason Oikonomidis, Antonis Argyros, and Tae-Kyun Kim. 2018. Depth-Based 3D Hand Pose Estimation: From Current Achievements to Future Goals. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.